WP4 Template 3 - Simulation of an impacting oscillator using an event based timestepping scheme implemented in Matlab

P. T. Piiroinen

Version 1.0
April 14, 2006

1 Description of the physical problem: A periodically forced pendulum with impacts

2 Definition of a general abstract class of NSDS: Impacting NSDS

In this section, we provide a short description of a general abstract class of Non Smooth Dynamical system (NSDS): Impacting NSDS. More generally, we will try to present this type of system in a more general framework where the NSDS consists of:

1. a dynamical system with boundary conditions in terms of state variables,
2. a set of input/output variables and their relations with the state variables,
3. a set of nonsmooth laws which rely on the input/output variable.

2.1 Dynamical system and Boundary conditions

General non-linear case: second order system. The equation of motion of a mechnical system may be stated as

$$M(q)\ddot{q} + F(\dot{q}, q, t) = F_{ext}(\dot{q}, q, t) \tag{1}$$

where

- $q \in I\!R^n$ is the generalized coordinates vector. The dimension n of this vector is called the number of degree of freedom of the system. The first and the second time derivative of q, denoted $\dot{q}$ and $\ddot{q}$, are usually called the velocity and the acceleration of the system.
- $M(q) : I\!R^n \mapsto \mathcal{M}^{n\times n}$ is the inertia term
- $F(\dot{q}, q, t) : I\!R^n \times I\!R^n \times I\!R \mapsto I\!R^n$ is the internal force of the system,
- $F_{ext}(t) : I\!R \mapsto I\!R^n$ is the given external force,

General non-linear case: first order system. The equations of motion (1) may be reformulated in terms of the state vector as

$$M(q)\dot{q} = f(q, t) \tag{2}$$

which is a classical order one formulation of an ordinary differential equation (ODE), and where

- $q \in I\!R^{2n}$ is the generalized coordinates vector.

- $M(q) : I\!R^{2n} \mapsto \mathcal{M}^{2n \times 2n}$ is the inertia term
- $f(q,t) : I\!R^{2n} \times I\!R \mapsto I\!R^{2n}$

In a general way, we can denote the state of the system as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad \text{or} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \\ \tau(t) \end{bmatrix} \tag{3}$$

which are vectors of dimension $2 \times n$ and $2 \times n + 1$, respectively. The second form is used when a non-autonomous systems is transfered to an autonomous.

Boundary conditions

The boundary conditions for an Initial Value Problem (IVP) are given by

$$t_0 \in I\!R, \quad x(t_0) = x_0 \in I\!R^{2n}. \tag{4}$$

2.2 Relation between constrained variables and state variables

Formal Case. In a general way, the dynamical system is completed by a set of nonsmooth laws which do not directly concern the state vector. The set of such variables, denoted y, on which we apply the constraints, depends, in a very general way, of the state vector x and the time t:

$$y = h(x,t). \tag{5}$$

Dynamical system In the dynamical systems the structure of these relations is very particular and we assume that they can be written as

$$\begin{aligned} y &= h(x), & (6)\\ \dot{y} &= \langle \nabla h(x), f(x) \rangle, & (7)\\ \ddot{y} &= \langle \nabla \langle \nabla h(x), f(x) \rangle, f(x) \rangle. & (8) \end{aligned}$$

2.3 Definition of the Non Smooth Law between constrained variables

Several kind of nonsmooth laws may be formulated for dynamical system. For the purpose of this template, we define just the unilateral contact law and the impact law.

The Newton impact law may formulated as follows :

$$\text{if } y(t) = 0, \quad x(t^+) = x(t^-) + e(x)\dot{y}(t^-) = x(t^-) + e(x) \left\langle \nabla h(x(t^-)), f(x(t^-)) \right\rangle \tag{9}$$

3 The formalization of the forced impacting oscillator into the class of impacting NSDS

We assume that the system of a forced impacting pendulum belongs to the abstract class of the NSDS.

3.1 Dynamical system

From physical data we can find all terms which define a second order NSDS (compare with (1)). In our special case we get for the free flight

$$\ddot{\theta} + \kappa\dot{\theta} + \frac{g_e}{L}\sin(\theta) = \frac{A\omega^2}{L}\sin(\omega t), \tag{10}$$

where θ is the angle of the pendulum, L is its effective length, κ is the linear damping coefficient, g_e is the effective gravity, ω is the forcing frequency, A is the forcing amplitude, and t is the time. Following (1) we have that

$$\begin{aligned} x &= \theta, && (11)\\ M(x,t) &= 1, && (12)\\ F(x,x',\tau) &= \frac{2\beta}{\eta}x' + \frac{1}{4\eta^2}\sin(x), && (13)\\ F_{ext}(x,\tau) &= \alpha\cos(x)\sin(\tau). && (14) \end{aligned}$$

In what follows we will instead use the ODE formulation (2) of the equations of motion instead. By letting $\tau = \omega t$ and $'$ denote $\frac{d}{d\tau}$ the nondimensionalised equations of motion become

$$\theta'' + \frac{2\beta}{\eta}\theta' + \frac{1}{4\eta^2}\sin(\theta) = \alpha\cos(\theta)\sin(\tau) \tag{15}$$

and thus if we let

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \\ \tau \end{bmatrix} \tag{16}$$

we have

$$M(x,t)\dot{x} = f(x), \tag{17}$$

where

$$\begin{aligned} M(x,t) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} && (18)\\ f(x) &= \begin{bmatrix} x_2 \\ -\frac{2\beta}{\eta}x' - \frac{1}{4\eta^2}\sin(x) + \alpha\cos(x)\sin(\tau) \\ 1 \end{bmatrix}. && (19) \end{aligned}$$

3.2 relations

The unilateral constraint requires that

$$h = \theta - \hat{\theta} = x_1 - \hat{\theta} \geq 0, \tag{20}$$

so we identify the terms of the equation (6)-(8) to be

$$\begin{aligned} y &= x_1 - \hat{\theta}, && (21)\\ y' &= \langle \nabla h(x), f(x,\tau)\rangle = x_2 && (22)\\ y'' &= \langle \nabla \langle \nabla h(x), f(x,\tau)\rangle, f(x,\tau)\rangle = -\frac{2\beta}{\eta}x' - \frac{1}{4\eta^2}\sin(x) + \alpha\cos(x)\sin(\tau). && (23) \end{aligned}$$

3.3 Nonsmooth laws

The Newton impact law at impact is given by

$$\text{if } y = 0, \quad \text{then} \quad x_2(\tau^+) = x_2(\tau^-) + e(x)\dot{y}(\tau^-) = -r x_2(\tau^-), \tag{24}$$

where

$$e(x) = \begin{pmatrix} 0 \\ -(1+r) \\ 0 \end{pmatrix}. \tag{25}$$

4 Description of the numerical strategy: Simulation using an event-driven sceheme in Matlab

4.1 Numerical implementation of the event driven scheme in Matlab

In this section we provide an idea how to use an event driven scheme to solve impact type problems (in Matlab).

4.1.1 Solvers and event detection

Nonlinear systems

For nonlinear dynamical systems any of the built-in adaptive integrators, such as `ode45` (4th order Runge-Kutta for nonstiff systems) or `ode15s` (uses backward differential formulas (BDFs) also known as Gear's method for stiff systems) can be used to solve the smooth dynamics. It is of course possible to write your own code for solving ODEs, which should be straight forward.

For nonlinear dynamical systems the built-in event detection routines that is integrated with the numerical integrators can be used to locate $y = 0$. For more grazing sensitive simulations, i.e. to make sure that low-velocity impacts are not lost, one can also look for $\dot{y} = \nabla h(x) f(x) = 0$. If a user-provided solver is used then the user also has include an event detection scheme. For this, e.g. standard bisection, secant, or Broyden's method (a variant of the secant method) can be used and with a specified toloreance.

Linear system

For linear dynamical systems any of the built in integrators (as for the nonlinear systems) can be used to solve the smooth dynamics. However, it is also possible to use the `expm` function which is the matrix exponential or in verys simple cases the actual solution can be given to the program for.

The same approach on event detection holds for linear systems as for nonlinear systems (see above).

Chatter and sticking/sliding

In impacting systems chatter and sticking are very common phenomena that often causes a lot of problems in event driven methods. To have sticking there are three condition that has to be fulfilled at impact. They are

$$
\begin{aligned}
y &= 0 \quad \text{(impact condition)} && (26)\\
\dot{y} &= 0 \quad \text{(grazing condition)} && (27)\\
\ddot{y} &\leq 0 \quad \text{(acceleration condition)} && (28)
\end{aligned}
$$

Sticking can be caused by either if the coefficient of restitution is zero or as the result of a chatter sequence. In chatter the system recognises infinite number of impacts in finite time. Obviously it is impossible to numerically solve for an infinite number of events, so the most common way to solve this is to set $\dot{y} = 0$ as the conditions (26) and (27) are fulfilled and

$$|\dot{y}| < C, \quad 0 < C << 1 \tag{29}$$

where C is a small and positive constant.

When the system is in sticking mode the condition to look for is **not** $y = 0$, since this is allready fulfilled, but instead we look for $\ddot{y} = 0$ which is exactly when the acceleration is changing sign and eventually the direction of the velocity away from the impacting surface.

4.1.2 Detection of and action at the impacting surface

In order to apply Newton's impact law the surface given by

$$y = h(x) = 0 \tag{30}$$

and possibly also

$$\dot{y} = \nabla h(x) f(x) = 0 \tag{31}$$

has to be locted with an appropriate method, as describe above in Sec. 4.1.1.

During the ODE integration the event detector will eventually find the exact time and position of an impact, i.e. when $y = 0$. At such a point the integrator stops, wherafter the impact law is applied and the integrator is again started, but now with the new inital conditions given by the impact law.

Chatter and sticking/sliding

When the motion fulfills (26), (28), and (29) the solver stops and $\dot{y}$ is set to 0 and the integration is restarted but now only looking for the event $\ddot{y} = 0$. To avoid drift away from the surface due to numerical errors tt is possible to make the surface attracting by adding the term

$$D \nabla h(x)^T h(x), \quad D > 0 \tag{32}$$

to the vector field. The constant should be used to make the time scales for the new term and the sticking vector field equal.

Another and a more recent idea is to use a local mapping att chattering. Such methods have been developed but not yet tested or included in a code.

4.2 Numerical Strategy

5 Exploitation of the results

For the case of Lagrangian NSDS, the state variable and the reaction force vs. time are the basic results that we want to export out of the algorithm. Various of energies may be also expected by the user.

6 First analysis for the architectural point of view

To be defined . . .